CONTENT MANAGEMENT SYSTEM

BACKGROUND

REFERENCE TO RELATED APPLICATIONS

The present disclosure is based on and claims the benefit of Provisional Application 60/433,264 filed December 13, 2002, the entire contents of which are herein incorporated by reference.

TECHNICAL FIELD

The present disclosure relates to content and, more specifically, to a content management system.

DESCRIPTION OF THE RELATED ART

Today, computer network security is a matter of the utmost importance. Networks may include a wide range of security tools to provide a level of network security. Even with the use of such security tools, network vulnerabilities and configuration problems may still pose a potentially costly security risk.

Vulnerabilities are technology faults that have been discovered. Configuration standards are instructions for implementing and auditing specific technologies. People can be used to correct vulnerabilities and configuration standards. Policies can be used to help people know what to do and to provide a system of checks to make sure that the treatment of vulnerabilities runs efficiently and effectively.

To better manage the treatment of vulnerabilities and configuration standards, corrective measures may be divided into discrete tasks that are then distributed to individuals. Detailed procedures for identifying tasks, distributing tasks, acknowledging tasks, and capturing completion of tasks may help companies attain an acceptable level of risk through a repeatable process.

SUMMARY

A method for monitoring technology information for vulnerabilities including an automated workflow process for detecting a vulnerability, researching the vulnerability and documenting the vulnerability within vulnerability data.

A method for monitoring technology information for configuration standards including an automated workflow process for initiating a configuration standard, researching the configuration standard and documenting the configuration standard within configuration standard data.

A method for developing configuration standards for use with an automated workflow process including initiating a content entry, researching the content entry, validating the content entry, approving the content entry and publishing the content entry to a database of approved configuration standards.

A method for updating content within a content management system using an automated workflow process, where content within the content management system is updated by a content update system that uses a pull methodology by allowing systems to obtain updated content when requested rather that pushing data onto the systems.

A method for creating policies for use within a content management system using an automated workflow process including initiating a content entry, researching the content entry, validating the content entry, approving the content entry and publishing the content entry to a database of approved policies.

An automated workflow system for monitoring technology information for vulnerabilities including a detector for detecting a vulnerability, a researcher for researching the vulnerability and a documenter for documenting the vulnerability within vulnerability data.

An automated workflow system for monitoring technology information for configuration standards including an initiator for initiating a configuration standard, a researcher for researching the configuration standard and a documenter for documenting the configuration standard within configuration standard data.

A system for developing configuration standards for use with an automated workflow system including an initiator to initiate a content entry, a researcher to research the content entry, a validator to validate the content entry, an approver to approve the

content entry and a publisher to publish the content entry to a database of approved configuration standards.

A system for updating content within a content management system using an automated workflow system including a content update system for updating the content within the content management system, where the content update system uses a pull methodology allowing systems to obtain updated content when requested rather that pushing data onto the systems.

A system for creating policies for use within a content management system using an automated workflow system, including an initiator for initiating a content entry, a researcher for researching the content entry, a validator for validating the content entry, an approver for approving the content entry and a publisher for publishing the content entry to a database of approved policies.

A computer system including a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for monitoring technology information for vulnerabilities, the method steps including detecting a vulnerability, researching the vulnerability and documenting the vulnerability within vulnerability data.

A computer system comprising a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for monitoring technology information for configuration standards including an automated workflow process for initiating a configuration standard, researching the configuration standard and documenting the configuration standard within configuration standard data.

A computer system comprising a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for developing configuration standards for use with an automated workflow process including initiating a content entry, researching the content entry, validating the content entry, approving the content entry and publishing the content entry to a database of approved configuration standards.

A computer system comprising a processor; and a program storage device readable by the computer system, embodying a program of instructions executable by the

processor to perform method steps for updating content within a content management system using an automated workflow process, where content within the content management system is updated by a content update system that uses a pull methodology by allowing systems to obtain updated content when requested rather that pushing data onto the systems.

A computer system comprising a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for creating policies for use within a content management system using an automated workflow process, including initiating a content entry, researching the content entry, validating the content entry, approving the content entry and publishing the content entry to a database of approved policies.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present disclosure and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

FIG. 1 shows a high-level view of the quality assurance process for vulnerabilities;

FIG. 2 shows a high-level view of the quality assurance process for Configuration Standards;

FIG. 3 shows a high-level view of the workflow for entering new policies into the CMS;

FIG. 4 shows a flow diagram for the introduction of new vulnerabilities into the CMS;

FIG. 5 shows a flow diagram detailing how a vulnerability is researched and documented;

FIG. 6 shows a flow diagram detailing how a vulnerability is validated;

FIG. 7 shows a flow diagram detailing how a vulnerability is approved and published; and

FIG. 8 illustrates an example of a computer system capable of implementing the

method and apparatus of the present disclosure.

DETAILED DESCRIPTION

In describing the preferred embodiments of the present disclosure illustrated in the drawings, specific terminology is employed for sake of clarity. However, the present disclosure is not intended to be limited to the specific terminology so selected, and it is to be understood that each specific element includes all technical equivalents which operate in a similar manner.

Vulnerabilities are technology faults that have been discovered. Configuration standards are instructions for implementing specific technologies. Vulnerabilities that go uncorrected can threaten network security by allowing an unauthorized person or program to access information technology systems, or assets, that are connected to the network. Configuration standards dictate how security features that protect network assets are configured. Poorly configured security features can also severely threaten network security.

Automated content management systems (CMS) are used to better manage the treatment of vulnerabilities and configuration standards that can threaten network security. According to an embodiment of the present disclosure, the CMS is a computer program, generally running on a computer, for example a network server, which organizes and manages the actions of individuals in their treatment of vulnerabilities and configuration standards. Individuals who use a CMS to manage the treatment of vulnerabilities and configuration standards are known as "users." Each user can be assigned one or more roles. A role dictates the types of tasks that may be assigned to an individual user. Roles can also be assigned to a responsibility group. A responsibility group is a category of users that share a particular skill set. Tasks that are assigned to a responsibility group can be completed by any member of that responsibility group.

The present disclosure relates to an automated CMS. According to an embodiment of the present disclosure, measures for correcting vulnerabilities and configuration standards are divided into discrete tasks that are then distributed to users according to their associated responsibility group. A task that has been completed by one user may then lead to a subsequent task being created for another user until the

vulnerability or configuration has been satisfactorily remedied. When one task relating to the remediation of a specific vulnerability or configuration standard is completed by one user and as a result a second task relating to the remediation of the same specific vulnerability or configuration standard is created and assigned to a second user, for simplicity, this scenario is herein referred to in terms of the vulnerability being sent or routed from the one user to the second user. This propagation of tasks from user to user may be referred to as a workflow. According to an embodiment of the present disclosure, the CMS provides an automated workflow where new tasks are automatically created and assigned to users and completed tasks may automatically trigger the creation of subsequent tasks.

According to an embodiment of the present disclosure, the CMS includes a quality assurance (QA) process. The QA process allows the CMS to manage tasks through the workflow to ensure that vulnerabilities and configuration standards are remedied with a repeatable high level of quality. The QA process associates roles with individual users.

During the process of working on tasks, users may generate content. Content can be text, computer code or anything else that may contribute to remediation of the vulnerability or configuration standard associated with the user's current task.

Users may implement corrections by creating new content or editing old content. When the user has completed a task and content has been changed, the CMS creates a new task for a user with a role of approver to review the changed content and potentially approve the changes made. According to embodiments of the present disclosure, there may be multiple approvers corresponding to multiple hierarchical approval levels. Changes made to content do not become effective until approved by a final approver. After content changes have been finally approved, the changed content is added to a content database. Subsequent tasks requiring access to the updated content will be able to pull the updated content off of the content database. If the changes are not approved, the changes are erased or stored for later editing and the content reverts to its prior state. In order to prevent multiple users from changing content at the same time, content may be locked while a user is currently working on a task and when the content is pending approval.

Each user may be assigned multiple tasks. Each user has a task list where all tasks assigned to that user are listed. The CMS assigns tasks to individual users or to a responsibility group and these tasks show up on the task lists of the appropriate users. The task list will also indicate the status of the tasks listed. A task has the status of open when the task is available to be completed by a user within the group the task is assigned to. A task has the status of personal when the task is currently being worked on by the user who's task list the task is listed on. A task has the status of locked when another user within the group is currently working on the task. The task list may also indicate the priority of the tasks listed. Priority is the level of importance of the task. For example, a task's priority may be high, medium, or low. The task list may also indicate the date the task was submitted to the CMS. The task list may also indicate the name of the task, the technology asset that the task affects, and/or the QA step the task is currently at. The QA step is an indication of how far along in the quality assurance process the vulnerability or configuration standard has come. When a task is referred to herein as being assigned to a user such as a reviewer, researcher, etc., it should be understood that the task may be assigned to a specific user or to a group of users with the specific roles of reviewer, researcher, etc.

A user may view a task listed on his or her task list. Viewing a task allows the user to see the content associated with the task. A user viewing a task may not make changes to the corresponding content. Other viewers can still access the task and its content even when a user is currently viewing that task. The user may also open the task. When the task is open, the user is permitted to make changes to the corresponding content, however, other users may not open the opened task.

The user may change the order in which tasks are displayed in the task list by the use of a filter. Filters may display tasks by content type. Content type indicates if the task relates to a vulnerability or a configuration standard. Filters may also display tasks by status or priority.

According to an embodiment of the present disclosure, users may be assigned a level of experience. For example, the level of experience may indicate how much experience the user has in dealing with assigned tasks. The experience level of a user will help the CMS to determine how many levels of review are required before finally

approving the content changes that user has made. For example, users with little experience may require more levels of review than more experienced users.

At each QA step, users may enter a reference name/number and a new technology name. The technology name and reference name/number identify what asset the vulnerability or configuration standard relates to. Changes made to names and references of assets are presented to an approver for approval and will not become effective until after final approval has been given. After final approval has been given, names and references will be added to the content database.

Embodiments of the present disclosure may use technology names that utilize a hierarchical structure to demonstrate the relationship between related assets. The technology name can include, for example, vendor name, product name, release number, minor release number, service pack number and/or other descriptive names. Vulnerabilities and configuration standards can relate to either a specific asset or a family of assets. When the vulnerability or configuration standard relates to a family of assets, the technology name used may be the technology name that includes all of the affected assets. For example, if a vulnerability relates to every release number for a given product name, that vulnerability may be identified with the vendor name and the product name. If a vulnerability relates only to a specific minor release number, the technology name may be the vendor name, the product name, the release number and the minor release number. Remedial steps taken for a family of assets may be applied to all assets within that family.

Users who have opened a task may make additions to a workflow comment field that is part of the vulnerability or configuration standard's content. Workflow comments may be displayed along with content when a task is opened by a user. Workflow comments may be displayed with the most recent additions appearing first.

According to an embodiment of the present disclosure, each user may have an associated user account. The user account is maintained by an administrator of the CMS. The user account may store information such as the user's company name, login name and a password conforming to set password standards. Users login to the CMS in order to gain access to their task lists.

The CMS captures and stores CMS usage data. Data relating to the times users

log in and out is recorded. The date vulnerabilities and configuration standards are submitted to the CMS is also recorded. The length of time for which the vulnerability or configuration standard is in the CMS may also be recorded. According to an embodiment of the present disclosure, this length of time is taken from the time the first task relating to the vulnerability or configuration standard is initiated to the time the task of final approval is completed. This information is particularly recorded for high priority vulnerabilities and configuration standards. Length of time data may also be recorded for all discrete tasks relating to all remediation. Recorded data may then be used to generate metrics such as a user activity report.

Users' accounts may be inactivated by the CMS administrator. When a user account is inactivated, all open tasks associated with that user will revert back to the user's group or will be reassigned.

FIG. 1 shows a high-level view of the QA process relating to the remediation of vulnerabilities. The diagram specifies the QA step as well as the role of the user who may be assigned the task relating to that QA step. When a task is assigned to a user, that task will appear in the task list of that user. When vulnerabilities are sent to another user, a new task is created in the task list of that other user and the original task is completed. The first task, according to this embodiment of the present disclosure, is assigned to a user with a role of vulnerability initiator. The vulnerability initiator can initiate a new vulnerability (step S1). The vulnerability initiator may create content related to the new vulnerability. For example, the content may include a description of the vulnerability. In order to prevent unnecessary delay in the automated CMS, users may only have a task open for a set amount of time. For example, according to an embodiment of the present disclosure, vulnerability content may only be open for a period less than 48 hours or the vulnerability is unlocked and changes made to the content are lost. The user may be warned of this fact after having the vulnerability open for 24 hours.

After the vulnerability initiator initiates the new vulnerability (step S1), thereby completing the assigned task, a user with the role of vulnerability reviewer performs an initial review (step S2). Vulnerabilities to be reviewed will appear in the task list of the vulnerability reviewer who will review the vulnerability content. The initial reviewer may reject the vulnerability if, for example, the vulnerability already exists in the CMS or

is known to not be a valid vulnerability. For example, a vulnerability may be known to not be a valid vulnerability if, for example, the same suspected vulnerability has in the past been rejected. If the vulnerability is rejected, the vulnerability may be sent to the task list of a vulnerability final approver for final rejection (step S8). Final rejection may end the remediation of the vulnerability. The vulnerability reviewer may also approve the vulnerability (step S2) thereby completing the assigned task. Approved vulnerabilities are then assigned to a user or group of users with a role of vulnerability researcher. If the task is assigned to a group of vulnerability researchers, the task may appear in each user's task list in the group until one user in the group opens the task at which point the other users in the group can no longer open the task. If the task is assigned to a specific user, only that user may open the task. The user who first opens the task may research the vulnerability and update the content accordingly (step S3). The researcher will either mark the vulnerability for rejection and send it to the final approver (step S8), send the vulnerability to a consultant (step S4), send the updated vulnerability content to a vulnerability validator (step S6) or mark the vulnerability with a pre-alert flag if the researcher believes the vulnerability to be a major vulnerability. Vulnerabilities may be deemed major, for example, when they affect a major asset, the vulnerability has not yet been recognized by the vendor and no patch to correct the vulnerability exists or the vulnerability is serious and affects a variety of non-major assets. When the researcher, or a validator sends the vulnerability to a consultant, the consultant will assist in the research and validation process (step S4). The consultant can edit the vulnerability content and then send it back to the researcher for further research (step S3). The consultant may be any user affiliated with the management of the information technology to be managed or an individual not affiliated with the information technology to be managed. When the researcher marks the vulnerability with a pre-alert flag and submits the vulnerability back into the workflow, the final approver will receive the pre-alert in his or her task list (step S5). The final approver can approve the pre-alert or reject the pre-alert. In either case, the vulnerability is sent to the task list of the vulnerability researcher. When the vulnerability researcher determines that research is completed, the vulnerability is sent to the vulnerability validator (step S6). The vulnerability validator will validate the vulnerability content. This involves either,

marking the vulnerability for rejection, sending the vulnerability to a consultant for consultation (step S4), returning the vulnerability to the researcher (step S3) to continue research or validating the vulnerability content. When the vulnerability validator validates the vulnerability content (step S6), the vulnerability is moved to the vulnerability technical editor's task list (step S7). The technical editor will edit the vulnerability content for format and clarity. The vulnerability is then sent to the task list of the vulnerability final approver (step S8). The vulnerability final approver will perform the final approval step where he or she has the ability to either reject the vulnerability, return the vulnerability to the researcher (step S3) to continue research or approve the vulnerability content. Vulnerability content that has been approved by the vulnerability final approver is added to the content database.

FIG. 2 shows a high-level view of the QA process for remediation of configuration standards. When one user sends a configuration standard to another user, thereby completing a task, a new task is created in the task list of that other user. The configuration standard initiator initiates a new configuration standard (step S11). While the configuration standard is being created, the configuration standard will be locked and no other users may open the configuration standard content. After the configuration standard has been initiated, it is sent to a configuration standard reviewer (step S12). The configuration standard reviewer performs an initial review of the configuration standard. The configuration standard reviewer may either assign the configuration standard to a research group or an individual researcher (step S13). The initial reviewer can also reject the configuration standard if, for example, it already exists in the CMS or is known to not be a valid configuration standard. The configuration standard researcher performs research on the configuration standard (step S13). The configuration standard researcher has the ability to either mark the configuration standard for rejection and have the configuration standard presented to the final approver for rejection (step S17), send the configuration standard content to a consultant (step S14) or update the configuration standard content and send it to the configuration standard validator (step S15). The consultant may receive an email when the task enters his or her task list. The configuration standard consultant may assist in the research and validation of the configuration standard (step S14). The consultant can edit the configuration standard

content and then send it back to the researcher (step S13) or validator (step S14) depending on who sent it. If the consultant does not open the task within five days, the task will be returned to the researcher or validator who sent it. In step S15, the configuration standard validator can either mark the configuration standard for rejection and have the configuration standard sent to the configuration standard final approver for final approval (step S17), send the configuration standard to consultant (step S14), return the configuration standard to the researcher (step S13) to continue the research or validate the configuration standard content and have it sent to the configuration standard technical editor (step S16). In step S16, the configuration standard technical editor edits the configuration standard content for format and clarity and then sends it to the configuration standard final approver. In step S17, the configuration standard final approver either rejects the configuration standard, returns it to the researcher (step S13) or validator (step S15) or approves the configuration standard content. Approved configuration standard content is added to the content database.

Policies are text documents that may be used to regulate the behavior of users. FIG. 3 shows a high-level view of the workflow for entering new policies into the CMS. During initiation (step S21), a user initiates a new content entry using a graphic user interface and the content is assigned to a user who is certified for handling the content type. This user will research the content (step S22) and may either reject it, sending it to the final approver (step S27), or send it to be validated (step S23). At the validation step S23, the validator can accept the content and forward it to a technical editor for editing (step S24). The validator can also reject the content and notify the final approver (step S28). If information is missing, the validator can return the content to the researcher for further research (step S22). During edit (step S24), the technical editor edits the content for format and clarity and sends it to an approval queue (step S25). The approval queue may be, for example, the task list of the approver. At the approval step S25, the approver can accept, reject or rout the submission back to the validator for additional information. If rejected, the submission is saved as not approved (step S29). If the approver has a question, the submission can be returned to the validator for further validation (step S23). If accepted by the approver, the content is sent to publishing (step S26). During publishing a research team can perform a final check prior to publication

and then the content can be published to the content database (step S30).

FIG. 4 shows a flow diagram for introducing of new vulnerabilities into the CMS. During web monitoring and research (step S31), a research team monitors internet newsgroups, mailing lists and alert services to obtain information about new vulnerabilities. When a potential vulnerability is recognized, a researcher submits vulnerability content to a content development initiation queue (step S32). The content development initiation queue may be, for example, part of the task list of the vulnerability content manager. If the vulnerability content manager deems the potential vulnerability to be major, a pre-alert notification is immediately issued. Vulnerabilities may be deemed major, for example, when they affect a major asset, the vulnerability has not yet been recognized by the vendor and no patch to correct the vulnerability exists or the vulnerability is serious and affects a variety of non-major assets. A content manager assigns each new vulnerability to an appropriate researcher for research. The researcher may analyze, test and/or document the potential vulnerability to verify that the vulnerability exists (step S33). If the vulnerability is deemed to be real, the researcher may add a unique description of the vulnerability to the vulnerability content. The researcher may also assign values to indicate the impact the vulnerability may have on assets, the popularity of the vulnerability and/or the complexity of the technique(s) necessary for exploiting the vulnerability. The researcher then may document any vendor patches for the vulnerability and/or any other countermeasures for mitigating the risk in the vulnerability content. The vulnerability is then sent to a validator, who reviews the vulnerability content for accuracy and completeness (step S34). A technical editor may then review the vulnerability content to ensure that the language is clear and that the style complies with set standards (step S35). The vulnerability content manager may then review the vulnerability content to ensure the information is accurate and complete (step S36). An approver can then perform a quality assurance check and then rout the vulnerability content back to the vulnerability content manager for publication to the content database (step S37).

FIG. 5 shows a flow diagram providing more detail how a vulnerability is researched and documented as performed in step S33 of Fig. 4. After the vulnerability content manager assigns a vulnerability to the task list or queue of a researcher (step

S41), the researcher checks the vulnerability database to see if the vulnerability has already been reported (step S42). The researcher may review the vulnerability and attempt to find additional sources establishing the same vulnerability (step S43). If a second source for the vulnerability can be found (yes, step S43) the researcher researches and documents the vulnerability (step S44). The researcher will then submit the vulnerability for review (step S45) and the vulnerability will proceed to validation (step S60). If no second source can be found (no, step S43), the researcher will attempt to verify the vulnerability with the vendor or test for the vulnerability (step S46). If the vulnerability can be verified (yes, step S47), the vulnerability is documented in the vulnerability content (step S48), submitted for review (step S49) and sent for validation (step S60). If the vulnerability cannot be verified (no, step S47), the results of the search are noted in the content and the vulnerability is sent to the vulnerability content manager (step S50). The content manager can review the vulnerability content (step S51) and return it for further research (step S52) if he believes the unverified vulnerability can be verified (YES, step S54). In the alternative, the content manager can send the vulnerability content to a file for unverified vulnerabilities for later research (step S53) if he believes that the unverified vulnerability can not be verified with additional research (NO, step S54).

FIG. 6 shows a flow diagram providing more detail how a vulnerability can be validated and edited as performed in steps S34 and S35 of Fig. 4. The validator receives the vulnerability that has been sent for review in his or her task list (step S62). The validator assesses the nature of the vulnerability to determine the vulnerability's impact, popularity and simplicity of exploitation and may review any external references found by the researcher (step S63). If the validator determines that the vulnerability is not valid (no, step S64), the validator may enter comments into the vulnerability content and rout the vulnerability back to the vulnerability manager (step S65). If the validator determines that the vulnerability is valid (yes, step S64) the validator may determine if the information relating to the vulnerability is complete (step S66). If it is determined to be incomplete (no, step S66), comments may be entered into the vulnerability content and the vulnerability routed back to the researcher (step S67). If the vulnerability is determined to be complete (yes, step S66), the vulnerability may be routed (step S68) to

the vulnerability content manager for review (step S69). If the vulnerability content manager determines that the vulnerability is invalid (no, step S70) it can be sent to an unverified vulnerability file for later research (step S71). If it is determined that the vulnerability is valid (yes, step S70), the vulnerability content manager can determine if the information relating to the vulnerability is complete (step S72). If it is not complete (no, step S72), comments may be added to the vulnerability content and the vulnerability routed back to the researcher (step S73). If it is complete (yes, step S74), the vulnerability can be routed (step S74) to the technical editor for review (step S75). The technical editor may edit the vulnerability content for language and conformity with set standards and then route the vulnerability (step S76) to the vulnerability manager for approval and publication.

FIG. 7 shows a flow diagram providing more detail how a vulnerability is reviewed, approved and published as performed in steps S36 and S37 of Fig. 4. The vulnerability is received from the technical editor and reviewed by the vulnerability content manager (step S82). If for any reason the vulnerability is not acceptable (no, step S83), it can be routed back to the researcher, validator or technical editor for further research, validation and/or technical review (step S85). If the vulnerability is acceptable (yes, step S83) it can be routed (step S84) to the approver for review (step S86). If the approver finds the vulnerability to be unacceptable (no, step S87), the vulnerability is routed back to the vulnerability manager (step S88). If the approver finds the vulnerability to be acceptable (yes, step S87), the approver approves the vulnerability for publication (step S89) and sends the vulnerability to the vulnerability content manager for publication (step S90). The vulnerability content manager then publishes the vulnerability (step S91) to a vulnerability database.

FIG. 8 shows an example of a computer system which may implement the method and system of the present disclosure. The system and method of the present disclosure may be implemented in the form of a software application running on a computer system, for example, a mainframe, personal computer (PC), handheld computer, server, etc. The software application may be stored on a recording media locally accessible by the computer system and accessible via a hard wired or wireless connection to a network, for example, a local area network, or the Internet.

The computer system referred to generally as system 100 may include, for example, a central processing unit (CPU) 102, random access memory (RAM) 104, a printer interface 106, a display unit 108, a local area network (LAN) data transmission controller 110, a LAN interface 112, a network controller 114, an internal buss 116, and one or more input devices 118, for example, a keyboard, mouse etc. As shown, the system 100 may be connected to a data storage device, for example, a hard disk, 120 via a link 122.